

# Package: influence.SEM (via r-universe)

August 21, 2024

**Type** Package

**Title** Case Influence in Structural Equation Models

**Version** 2.3

**Date** 2022-05-06

**Author** Massimiliano Pastore & Gianmarco Altoe'

**Depends** lavaan

**Suggests** tcltk

**Maintainer** Massimiliano Pastore <massimiliano.pastore@unipd.it>

**Description** A set of tools for evaluating several measures of case influence for structural equation models.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-05-11 07:00:02 UTC

## Contents

bollen.loglik . . . . .	2
Deltachi . . . . .	3
explore.influence . . . . .	4
fitinfluence . . . . .	6
genCookDist . . . . .	7
Likedist . . . . .	9
parinfluence . . . . .	11
PDII . . . . .	12
Q . . . . .	13
sem.fitres . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

bollen.loglik      *Log-Likelihood of a sem model (Internal function).*

---

### Description

Internal function, called by [Likedist](#).

### Usage

```
bollen.loglik(N, S, Sigma)
```

### Arguments

N	Sample size.
S	Observed covariance matrix.
Sigma	Model fitted covariance matrix, $\Sigma(\theta)$ .

### Details

The log-likelihood is computed by the function [bollen.loglik](#) using the formula 4B2 described by Bollen (1989, pag. 135).

### Value

Returns the Log-likelihood.

### Author(s)

Massimiliano Pastore, Gianmarco Altoe'

### References

Bollen, K.A. (1989). *Structural Equations with latent Variables*. New York, NY: Wiley.

### See Also

[Likedist](#)

### Examples

```
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"
fit0 <- sem(model, data=PDII)
N <- fit0@Data@nobs[[1]]
S <- fit0@SampleStats@cov[[1]]
Sigma <- fitted(fit0)$cov
bollen.loglik(N,S,Sigma)
```

---

Deltachi	<i>Chi-square difference.</i>
----------	-------------------------------

---

**Description**

Quantifies case influence on overall model fit by change in the test statistic

$$\Delta_{\chi_i^2} = \chi^2 - \chi_{(i)}^2$$

where  $\chi^2$  and  $\chi_{(i)}^2$  are the test statistics obtained from original and deleted  $i$  samples.

**Usage**

```
Deltachi(model, data, ..., scaled = FALSE)
```

**Arguments**

model	A description of the user-specified model using the lavaan model syntax. See <a href="#">lavaan</a> for more information.
data	A data frame containing the observed variables used in the model. If any variables are declared as ordered factors, this function will treat them as ordinal variables.
...	Additional parameters for <a href="#">sem</a> function.
scaled	Logical, if TRUE the function uses the scaled $\chi^2$ (Rosseel, 2013).

**Value**

Returns a vector of  $\Delta_{\chi_i^2}$ .

**Note**

If for observation  $i$  model does not converge or yields a solution with negative estimated variances, the associated value of  $\Delta_{\chi_i^2}$  is set to NA.

This function is a particular case of [fitinfluence](#), see example below.

**Author(s)**

Massimiliano Pastore

**References**

- Pek, J., MacCallum, R.C. (2011). Sensitivity Analysis in Structural Equation Models: Cases and Their Influence. *Multivariate Behavioral Research*, 46, 202-228.
- Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48, 1-36.
- Rosseel, Y. (2022). The lavaan tutorial. URL: <https://lavaan.ugent.be/tutorial/>.

## Examples

```
## not run: this example take several minutes
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"

# fit0 <- sem(model, data=PDII)
# Dchi <- Deltachi(model,data=PDII)
# plot(Dchi,pch=19,xlab="observations",ylab="Delta chisquare")

## not run: this example take several minutes
## an example in which the deletion of a case yelds a solution
## with negative estimated variances
model <- "
  F1 =~ x1+x2+x3
  F2 =~ y1+y2+y3+y4
  F3 =~ y5+y6+y7+y8
"

# fit0 <- sem(model, data=PDII)
# Dchi <- Deltachi(model,data=PDII)
# plot(Dchi,pch=19,xlab="observations",ylab="Delta chisquare",main="Deltachi function")

## the case that produces negative estimated variances
# sem(model,data=PDII[~which(is.na(Dchi)),])

## same results
# Dchi <- fitinfluence("chisq",model,data=PDII)$Dind$chisq
# plot(Dchi,pch=19,xlab="observations",ylab="Delta chisquare",main="fitinfluence function")
```

---

explore.influence      *Explores case influence.*

---

## Description

It explores case influence. Cases with extreme values of the considered measure of influence are reported. Extreme values are determined using the boxplot criterion (Tukey, 1977) or user-defined cut-offs. Cases for which deletion leads to a model that does not converge or yelds a solution with negative estimated variances are also reported. In addition, explore.influence provides a graphical representation of case influence.

## Usage

```
explore.influence(x, cut.offsets = 'default',
                 plot = 'TRUE', cook = 'FALSE', ...)
```

**Arguments**

x	A vector containing the influence of each case as returned by <a href="#">Deltachi</a> , <a href="#">fitinfluence</a> , <a href="#">genCookDist</a> , <a href="#">Likedist</a> or <a href="#">parinfluence</a> functions.
cut.off.s	A vector of two numeric elements containing the lower and the upper cut-offs to be considered. If default, the cut-offs are calculated according to the boxplot criterion for outliers (see also, cook).
plot	If TRUE (the default) a graphical representation of case influence is given.
cook	If TRUE, x is interpreted as a vector containing Cook's distances, and so the lower cut-off is forced to be greater or equal to zero.
...	Additional parameters for plot function.

**Value**

A list with the following components:

n	number of cases.
cook	logical, indicating if x is treated as a vector of Cook's distances.
cut.low	the lower cut-off.
cut.upp	the upper cut-off.
not.allowed	a vector containing cases with negative variance or not converging models.
less.cut.low	a vector containing cases with influence value less than the lower cut-off.
greater.cut.low	a vector containing cases with influence value greater than the upper cut-off.

**Author(s)**

Gianmarco Altoe'

**References**

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.

**Examples**

```
data("PDII")
model <- "
F1 =~ y1+y2+y3+y4
"
fit0 <- sem(model, data=PDII, std.lv=TRUE)
## not run
# gCD <- genCookDist(model, data=PDII, std.lv=TRUE)
# explore.influence(gCD, cook=TRUE)

##
## not run: this example take several minutes
model <- "
F1 =~ x1+x2+x3
```

```

F2 =~ y1+y2+y3+y4
F3 =~ y5+y6+y7+y8
"

# fit0 <- sem(model, data=PDII)
# FI <- fitinfluence('rmsea',model,PDII)
# explore.influence(FI)

```

---

fitinfluence	<i>Case influence on model fit.</i>
--------------	-------------------------------------

---

### Description

This function evaluate the case's effect on a user-defined fit index.

### Usage

```
fitinfluence(index, model, data, ...)
```

### Arguments

index	A model fit index.
model	A description of the user-specified model using the lavaan model syntax. See <a href="#">lavaan</a> for more information.
data	A data frame containing the observed variables used in the model. If any variables are declared as ordered factors, this function will treat them as ordinal variables.
...	Additional parameters for <a href="#">sem</a> function.

### Details

For each case evaluate the influence on one or more fit indices: the difference between the chosen fit index calculated for the SEM target model  $M$  and the same index computed for the SEM model  $M_{(i)}$  excluding case  $i$ .

### Value

Returns a list:

Dind	a data.frame of case influence.
Oind	observed fit indices.

### Note

If for observation  $i$  model does not converge or yields a solution with negative estimated variances, the associated value of influence is set to NA.

**Author(s)**

Massimiliano Pastore

**References**

Pek, J., MacCallum, R.C. (2011). Sensitivity Analysis in Structural Equation Models: Cases and Their Influence. *Multivariate Behavioral Research*, 46, 202-228.

**Examples**

```
## not run: this example take several minutes
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"

# fit0 <- sem(model, data=PDII)
# FI <- fitinfluence("cfi",model,data=PDII)
# plot(FI$Dind,pch=19)

## not run: this example take several minutes
## an example in which the deletion of a case yields a solution
## with negative estimated variances
model <- "
  F1 =~ x1+x2+x3
  F2 =~ y1+y2+y3+y4
  F3 =~ y5+y6+y7+y8
"

# fit0 <- sem(model, data=PDII)
# FI <- fitinfluence(c("tli","rmsea"),model,PDII)
# explore.influence(FI$Dind$tli)
# explore.influence(FI$Dind$rmsea)
```

---

genCookDist

*Generalized Cook Distance.*


---

**Description**

Case influence on a vector of parameters may be quantified by generalized Cook's Distance (*gCD*; Cook 1977, 1986):

$$gCD_i = (\hat{\theta} - \hat{\theta}_{(i)})'_a \hat{\Sigma}(\hat{\theta}_{(i)})^{-1} (\hat{\theta} - \hat{\theta}_{(i)})$$

where  $\hat{\theta}$  and  $\hat{\theta}_{(i)}$  are  $l \times 1$  vectors of parameter estimates obtained from the original and delete  $i$  samples, and  ${}_a \hat{\Sigma}(\hat{\theta}_{(i)})$  is the estimated asymptotic covariance matrix of the parameter estimates obtained from reduced sample.

**Usage**

```
genCookDist(model, data, ...)
```

**Arguments**

model	A description of the user-specified model using the lavaan model syntax. See <a href="#">lavaan</a> for more information.
data	A data frame containing the observed variables used in the model. If any variables are declared as ordered factors, this function will treat them as ordinal variables.
...	Additional parameters for <a href="#">sem</a> function.

**Value**

Returns a vector of  $gCD_i$ .

**Note**

If for observation  $i$  model does not converge or yields a solution with negative estimated variances, the associated value of  $gCD_i$  is set to NA.

**Author(s)**

Massimiliano Pastore

**References**

Cook, R.D. (1977). Detection of influential observations in linear regression. *Technometrics*, 19, 15-18.

Cook, R.D. (1986). Assessment of local influence. *Journal of the Royal Statistical Society B*, 48, 133-169.

Pek, J., MacCallum, R.C. (2011). Sensitivity Analysis in Structural Equation Models: Cases and Their Influence. *Multivariate Behavioral Research*, 46, 202-228.

**Examples**

```
## not run: this example take several minutes
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"
# fit0 <- sem(model, data=PDII)
# gCD <- genCookDist(model,data=PDII)
# plot(gCD,pch=19,xlab="observations",ylab="Cook distance")

## not run: this example take several minutes
## an example in which the deletion of a case produces solution
## with negative estimated variances
model <- "
```

```

F1 =~ x1+x2+x3
F2 =~ y1+y2+y3+y4
F3 =~ y5+y6+y7+y8
"

# fit0 <- sem(model, data=PDII)
# gCD <- genCookDist(model,data=PDII)
# plot(gCD,pch=19,xlab="observations",ylab="Cook distance")

```

Likedist

*Likelihood Distance.***Description**

A general model-based measure of case influence on model fit is likelihood distance (Cook, 1977, 1986; Cook & Weisberg, 1982) defined as

$$LD_i = 2[L(\hat{\theta}) - L(\hat{\theta}_{(i)})]$$

where  $\hat{\theta}$  and  $\hat{\theta}_{(i)}$  are the  $k \times 1$  vectors of estimated model parameters on the original and deleted  $i$  samples, respectively, where  $i = 1, \dots, N$ . The subscript  $(i)$  indicates that the estimate was computed on the sample excluding case  $i$ .  $L(\hat{\theta})$  and  $L(\hat{\theta}_{(i)})$  are the log-likelihoods based on the original and the deleted  $i$  samples, respectively.

**Usage**

```
Likedist(model, data, ...)
```

**Arguments**

model	A description of the user-specified model using the lavaan model syntax. See <a href="#">lavaan</a> for more information.
data	A data frame containing the observed variables used in the model. If any variables are declared as ordered factors, this function will treat them as ordinal variables.
...	Additional parameters for <a href="#">sem</a> function.

**Details**

The log-likelihoods  $L(\hat{\theta})$  and  $L(\hat{\theta}_{(i)})$  are computed by the function [bollen.loglik](#) using the formula 4B2 described by Bollen (1989, pag. 135).

The likelihood distance gives the amount by which the log-likelihood of the full data changes if one were to evaluate it at the reduced-data estimates. The important point is that  $L(\hat{\theta}_{(i)})$  is not the log-likelihood obtained by fitting the model to the reduced data set. It is obtained by evaluating the likelihood function based on the full data set (containing all  $n$  observations) at the reduced-data estimates (Schabenberger, 2005).

**Value**

Returns a vector of  $LD_i$ .

**Note**

If for observation  $i$  model does not converge or yields a solution with negative estimated variances, the associated value of  $LD_i$  is set to NA.

**Author(s)**

Massimiliano Pastore, Gianmarco Altoe'

**References**

- Bollen, K.A. (1989). *Structural Equations with latent Variables*. New York, NY: Wiley.
- Cook, R.D. (1977). Detection of influential observations in linear regression. *Technometrics*, 19, 15-18.
- Cook, R.D. (1986). Assessment of local influence. *Journal of the Royal Statistical Society B*, 48, 133-169.
- Cook, R.D., Weisberg, S. (1986). *Residuals and influence in regressions*. New York, NY: Chapman & Hall.
- Pek, J., MacCallum, R.C. (2011). Sensitivity Analysis in Structural Equation Models: Cases and Their Influence. *Multivariate Behavioral Research*, 46, 202-228.
- Schabenberger, O. (2005). Mixed model influence diagnostics. In *SUGI*, 29, 189-29. SAS institute Inc, Cary, NC.

**See Also**

[bollen.loglik](#)

**Examples**

```
## not run: this example take several minutes
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"
# fit0 <- sem(model, data=PDII)
# LD <- Likedist(model, data=PDII)
# plot(LD, pch=19, xlab="observations", ylab="Likelihood distances")

## not run: this example take several minutes
## an example in which the deletion of a case yields a solution
## with negative estimated variances
model <- "
  F1 =~ x1+x2+x3
  F2 =~ y1+y2+y3+y4
  F3 =~ y5+y6+y7+y8
"
```

```
# fit0 <- sem(model, data=PDII)
# LD <- Likedist(model, data=PDII)
# plot(LD, pch=19, xlab="observations", ylab="Likelihood distances")
```

---

parinfluence                      *Case influence on model parameters.*

---

### Description

Computes direction of change in parameter estimates with

$$\Delta \hat{\theta}_{ji} = \frac{\hat{\theta}_j - \hat{\theta}_{j(i)}}{[VAR(\hat{\theta}_{j(i)})]^{1/2}}$$

where  $\hat{\theta}_j$  and  $\hat{\theta}_{j(i)}$  are the parameter estimates obtained from original and deleted  $i$  samples.

### Usage

```
parinfluence(parm, model, data, cook = FALSE, ...)
```

### Arguments

parm	Single parameter or vector of parameters.
model	A description of the user-specified model using the lavaan model syntax. See <a href="#">lavaan</a> for more information.
data	A data frame containing the observed variables used in the model. If any variables are declared as ordered factors, this function will treat them as ordinal variables.
cook	Logical, if TRUE returns generalized Cook's Distance computed as $[\Delta \hat{\theta}_{ji}]^2$ .
...	Additional parameters for <a href="#">sem</a> function.

### Value

Returns a list:

gCD	Generalized Cook's Distance, if cook=TRUE.
Dparm	Direction of change in parameter estimates.

### Note

If for observation  $i$  model does not converge or yields a solution with negative estimated variances or NA parameter values, the associated values of  $\Delta \hat{\theta}_{ji}$  are set to NA.

### Author(s)

Massimiliano Pastore

## References

Pek, J., MacCallum, R.C. (2011). Sensitivity Analysis in Structural Equation Models: Cases and Their Influence. *Multivariate Behavioral Research*, 46, 202-228.

## Examples

```
## not run: this example take several minutes
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"
# fit0 <- sem(model, data=PDII)
# PAR <- c("F1=~y2", "F1=~y3", "F1=~y4")
# LY <- parinfluence(PAR,model,PDII)
# str(LY)
# explore.influence(LY$Dparm[,1])

## not run: this example take several minutes
## an example in which the deletion of a case yields a solution
## with negative estimated variances
model <- "
  F1 =~ x1+x2+x3
  F2 =~ y1+y2+y3+y4
  F3 =~ y5+y6+y7+y8
"
# fit0 <- sem(model, data=PDII)
# PAR <- c("F2=~y2", "F2=~y3", "F2=~y4")
# LY <- parinfluence(PAR,model,PDII)

## not run: this example take several minutes
## dealing with ordinal data
data(Q)
model <- "
  F1 =~ it1+it2+it3+it4+it5+it6+it7+it8+it9+it10
"
# fit0 <- sem(model, data=Q, ordered=colnames(Q))
# LY <- parinfluence("F1=~it4",model,Q,ordered=colnames(Q))
# explore.influence(LY$Dparm[,1])
```

## Description

Simulated data set from covariance matrix reported in Bollen (1989).

**Usage**

```
data(PDII)
```

**Format**

This data frame contains 75 obs. of 11 variables:

- x1: num, gross national product per capita.
- x2: num, consumption per capita.
- x3: num, percentage of the labor force in industrial occupations.
- y1: num, freedom of the press in 1960.
- y2: num, freedom of group opposition in 1960.
- y3: num, fairness of elections in 1960.
- y4: num, elective nature and effectiveness of the legislative body in 1960.
- y5: num, freedom of the press in 1965.
- y6: num, freedom of group opposition in 1965.
- y7: num, fairness of elections in 1965.
- y8: num, elective nature and effectiveness of the legislative body in 1965.

**References**

Bollen, K.A. (1989). *Structural Equations with latent Variables*. New York, NY: Wiley.

**Examples**

```
data(PDII)
```

---

Q

*Simulated data set.*

---

**Description**

Simulated data set.

**Usage**

```
data(Q)
```

**Format**

This data frame contains 919 obs. of 10 ordinal discrete variables.

**Examples**

```
data(Q)
```

---

`sem.fitres`*Fitted values and residuals*

---

**Description**

It calculates the expected values and the residuals of a sem model.

**Usage**

```
sem.fitres(object)
obs.fitres(object)
lat.fitres(object)
```

**Arguments**

`object`            An object of class lavaan.

**Details**

The main function, `sem.fitres()`, calls one of the other two routines depending on the type of the model. If model does not contain latent variables, `sem.fitres()` calls the function `obs.fitres()`, otherwise calls the function `lat.fitres()`.

The functions `obs.fitres()` and `lat.fitres()` are internal functions, do not use it directly.

**Value**

Returns a data frame containing: 1) The observed model variables; 2) The expected values on dependent variables (indicated with `hat.`); 3) The residuals on dependent variables (indicated with `e.`)

**Note**

In order to compute more interpretable fitted values and residuals, model is forced to have `meansstructure = TRUE` and `std.lv = TRUE`.

**Author(s)**

Massimiliano Pastore

**Examples**

```
data("PDII")
model <- "
  F1 =~ y1+y2+y3+y4
"

fit0 <- sem(model, data=PDII)
out <- sem.fitres(fit0)
```

```
head(out)

par(mfrow=c(2,2))
plot(e.y1~hat.y1,data=out)
plot(e.y2~hat.y2,data=out)
plot(e.y3~hat.y3,data=out)
plot(e.y4~hat.y4,data=out)

qqnorm(out$.y1); qqline(out$.y1)
qqnorm(out$.y2); qqline(out$.y2)
qqnorm(out$.y3); qqline(out$.y3)
qqnorm(out$.y4); qqline(out$.y4)
```

# Index

## \* datasets

PDII, [12](#)

Q, [13](#)

## \* graphics

explore.influence, [4](#)

## \* multivariate

Deltachi, [3](#)

fitinfluence, [6](#)

genCookDist, [7](#)

Likedist, [9](#)

parinfluence, [11](#)

## \* utility

bollen.loglik, [2](#)

sem.fitres, [14](#)

bollen.loglik, [2](#), [2](#), [9](#), [10](#)

Deltachi, [3](#), [5](#)

explore.influence, [4](#)

fitinfluence, [3](#), [5](#), [6](#)

genCookDist, [5](#), [7](#)

lat.fitres (sem.fitres), [14](#)

lavaan, [3](#), [6](#), [8](#), [9](#), [11](#)

Likedist, [2](#), [5](#), [9](#)

obs.fitres (sem.fitres), [14](#)

parinfluence, [5](#), [11](#)

PDII, [12](#)

Q, [13](#)

sem, [3](#), [6](#), [8](#), [9](#), [11](#)

sem.fitres, [14](#)